Electronic Version
Stylesheet Version v1.1.1

Description

SELF-CONTAINED PROCESSOR SUBSYSTEM AS COMPONENT FOR SYSTEM-ON-CHIP DESIGN

BACKGROUND OF INVENTION

[0001] The present invention relates to network processor devices, and particularly, a system and method for simplifying the design of complex System-on-Chip (SOC) implementations by providing a self-contained processor subsystem as a component for System-on-Chip design.

[0002] Current state-of-the-art of building SoCs requires the designer to, inter alia: a) assemble from basic components such as microprocessors, memories, basic I/O macros (e.g., a framer); b) model bus-contention between the different devices, and select appropriate bus structures; c) integrate all components during SoC hardware design; and d) integrate all components using custom software.

[0003] However, there are inherent problems with the state-of-the art SoC design methodologies, including, but not limited to: it is labor-intensive; it is error-prone; it requires highly-skilled designers familiar with a particular application domain; it demands high modeling overhead for bus modeling and/or contention on a common system bus; it requires hardware and software integration to provide such basic services as TCP/IP, InfiniBand, FibreChannel, iSCSI and other standardized protocols. An example of a successful SoC integration design approach has been implemented in the MPC8560 Integrated Communications Processor available from Motorola, Inc.

[0004] Other approaches to SoC design where multiple subsystems are integrated on a card or board exhibit problems due to component count which drives system cost, increased failure susceptibility, and the cost of high-interconnect multi-layer boards.

[0005] Figures 1 and 2 illustrate respective prior art implementations of a Network Processor chip 10 (Figure 1) and 15 (Figure 2) each including multiple processing cores 20, local memory, data memory, link memory, CPU, buffers, and PHY (network physical layer) interfaces. These are stand-alone NPU's (Network Processor Units) that do not connect to an "open" system bus via a common macro.

[0006] Figure 3 illustrates a prior art implementation of a Network Processor chip 30 including processors 40, a system bus, cache, and local memory connected through a "bridge" such as the PCI (Peripheral Components Interconnection bus) bridge to a local processor bus commonly used in today's systems.

[0007] It would thus be highly desirable to provide an SoC integrated circuit having a multiprocessor subsystem as component and further a self-contained multiprocessor subsystem having predefined functionality for implementation as an independent SoC component and further, provides multithreading capability.

[0008] Relevant references describing aspects of SoC processor and component design include:

[0009] U.S. Patent No. 6,331,977 describes a System on a chip (SOC) that contains a crossbar switch between several functional I/Os internal to the chip and number of external connection pins, where the number of pins is less than the number of internal I/Os.

[0010] U.S. Patent No. 6,262,594 describes an apparatus and method implementing a crossbar switch for configurable use of group of pads of a system on chip.

[0011] U.S. Patent No. 6,038,630 describes an apparatus and method implementing a crossbar switch for providing shared access control device for integrated system with multiple functional units accessing external structures over multiple data buses.

[0012] U.S. Patent application No. US2002/0184419 describes an ASIC which enables use of different components for a system on a chip using a common bus system and describes wrappers for functional units with different speed and data width to achieve compatibility with a common bus.

[0013] U.S. Patent application No. US2002/0176402 describes an octagonal interconnection network for linking functional units on a SoC. The functional units on the interconnection network are organized as a ring and use several crossing data links coupling halfway components.

[0014] U.S. Patent application No. US2001/0042147 describes a system resource router for SOC interconnection, comprising two channel sockets with connect each data cache (D-cache) and instruction (I-cache). Also included are external data transfer initiators, two internal M-channel buses, and an M-channel controller to provide the interconnection.

[0015] U.S. Patent application No. US2002/0172197 describes a communication system connecting multiple transmitting and receiving devices via a crossbar switch embedded on a chip in a point-to-point fashion.

[0016] U.S. Patent application No. US2001/0047465 describes several variations of an invention providing a scalable architecture for a communication system (typically a SOC or ASIC) for minimizing total gates by dividing transmissions into individual transmission tasks, determining a computational complexity for each transmission task and computational complexity being based on the number of MIPS per circuit.

[0017] In the reference entitled "On-Chip Interconnects for Next Generation System-on-Chips" by A. Brinkmann, J. C. Niemann, I. Hehemann, D. Langen, M. Porrmann, and U. Ruckert, Conf. Proceedings of ASIC2003, Sept. 26-27, 2003, Rochester, New York, there is described an SoC architecture utilizing active switch boxes to connect processor cells for enabling packet network communications. This paper makes no mention or description of a processor core with multi-threading capability.

[0018] In the reference entitled "A Comparison of Five Different Multiprocessor SoC Bus Architectures" by Kyeong Keol Ryu, Eung Shin, and Vincent J. Mooney, Conf. proceedings of Euromicro Symposium on Digital System Design (DSS`01), Sept. 04-06, 2001, Warsaw, Poland, there is described Multiprocessor SoC bus architectures including Global Bus I Architecture (GBIA), Global Bus II Architecture (GBIIA), Bi-FIFO Bus Architecture (BFBA), Crossbar Switch Bus Architecture (CSBA), and CoreConnect Bus Architecture (CCBA).

[0019] None of the prior art configurations teach a processor core that comprises multiple sub-processors (thread groups) each with locally connecting SRAM or eDRAM in a multithreading configuration in order to improve processor performance and further SOC, ASIC, NP, or DSP integration.

Summary of Invention

[0020] It is an object of the present invention to provide a self-contained multiprocessor subsystem component functioning as a specially programmed component capable of performing multi-threading operations in an SoC integrated circuit.

[0021] In the present invention, a self-contained multiprocessor (MP) component includes sub-processor cores each containing local memory (e.g., SRAM) in order to enable a multi-threading processor core as a component in SoC design. Additionally included in the self-contained multiprocessor component is a local interconnect medium such as a crossbar switch (or similar type switch design) that connects to a single local processor bus of the SoC integrated circuit. The SoC IC may be configured as an advanced microprocessor, DSP (Digital Signal Processor), coprocessor, Hybrid ASIC, network processor (NP) or other like processor arrangement ASIC. Such an SoC integrated circuit having the self-contained multiprocessor subsystem component provides multi-threading capability whereby a sub-processor core (thread unit) operates independent of other threads by allowing program code and data from one context to remain independent from other contexts. The crossbar switch further enables communication with the rest of the chip via well-defined hardware and software interfaces.

[0022] In another aspect of the invention, the self-contained multiprocessor (MP) component as a component in SoC ASIC design is available as a ready made multi-threading processor core with appropriate software for a specific use. The MP component is connected to other components using a standardized interface such as a Processor Local Bus (PLB) adapter that bridges the local interconnect medium with a standardized ASIC methodology bus, such as CoreConnect-PLB bus, or any other on-chip bus or bus protocol.

[0023] The self-contained multiprocessor (MP) component providing multi-threading operations of the present invention not only improves processor speed, but reduces off-chip access times, significantly reduces cache latency, and improves instruction and data packet processing. Via a software polling technique that is easily reconfigurable, the processor core may be adapted for different communication protocols (Fibre Channel, Ethernet, IPsec, ATM, Ipv6, etc).

[0024] In another aspect of the invention, the multi-processor core includes polling software that enables the MP core to connect with the local processor bus and/or a common media interface MAC's such as Ethernet, Fibre Channel, iSCSI, etc. This enables more efficient data processing, reusable core design, protocol independent core design, and multiple numbers of system processing cores attached to a common processor bus for higher levels of SoC performance. When configured as SoC microprocessor designs the common bus-attached multi-processor enhances performance (faster speed, lower latency, drastically improved cache performance and/or even the elimination of off-chip cache or memory off-loads all together, except external storage, and requests). As processor speed increases (e.g., greater than 2 GHz - 10 GHz), the invention provides a most effective way to utilize the common microprocessor speed and memory cache bottleneck found in today's PC and workstation computer designs.

[0025] Advantageously, the present SoC design of the invention may be implemented for applications and uses including, but not limited to: IPSec VPN (Virtual Private Networks) tunneling engine; TCP/IP Offload Engine; Network processing for iSCSI; Multimedia processing, e.g., MPEG en/de-coding, sound/voice/video processing, Encryption engine, Compression/decompression engine, etc.

BRIEF DESCRIPTION OF DRAWINGS

[0026] Further features, aspects and advantages of the apparatus and methods of the present invention will become better understood with regard to the following description, appended claims, and the accompanying drawings where:

[0027] Figures 1 - 3 illustrate various prior art implementations of a Network Processor chip including multiple processing cores, memory and interfaces according to the prior art;

[0028] Figure 4 depicts an exemplary Processor Core used in a preferred embodiment of the present invention;

[0029] Figure 5 depicts an exemplary overview of a multiprocessor subsystem implementing functionality according to a preferred embodiment of the invention;

[0030] Figure 6 depicts a further embodiment of a Network Attached Processor employing the SoC subsystem of Figure 4(b) according to a second embodiment of the invention;

[0031] Figure 7 depicts an SoC employing processor-based subsystem according to a further embodiment of the invention;

[0032] Figure 8 depicts an SoC multiprocessor subsystem according to a further embodiment of the invention;

[0033] Figure 9 depicts a possible implementation of a bridge component provided in the system of Figure 8; and,

[0034] Figure 10 depicts one exemplary Network Processor arrangement 200 implementing the independent multiprocessor core 150' according to the invention.

DETAILED DESCRIPTION

[0035] Multiprocessor systems-on-a-chip consist of multiple instances of different components: (i) functional units; (ii) memory (including cache and main memory); and (iii) interconnection. Design choices include both the relative and absolute numbers of components, their particular features, and their placement with respect to each other.

[0036] Figure 4(b) depicts an exemplary self-contained processor-based subsystem 150 as a component for multiprocessor system on chip design according to the invention. In the example depicted in Figure 4(b) the self-contained processor-based subsystem 150 comprises a plurality of processor units 100, a shared memory such as provided by an SRAM memory 110 and a switch fabric 120. As shown in Figure 4(a), each processor unit 100 comprises a plurality of individual processor cores 125, for example, four (4) processing cores 125 comprising a processing unit or "Quad" 100 as depicted in Figure 4(a) with each processor core 125 comprising an execution unit or processor device, and connected with a common local (private) memory depicted as SRAM 130, e.g., providing 16KBytes of memory.

[0037] In one embodiment, the self-contained processor-based subsystem 150 depicted in Figure 4(b) is based on a multithreaded architecture chip design developed by the assignee of the present invention International Business Machines Corporation. (IBM), referred to herein as "Cyclops" and described in detail in the reference to C.J. Georgiou, et al. entitled "A programmable scalable platform for next generation networking," Proceedings of Workshop on Network Processors, Feb. 8-9, 2002, Anaheim, CA. A single Cyclops chip may comprise a large number (typically hundreds) of simple thread execution units, each one simultaneously executing an independent stream of instructions. The performance of each individual thread is such that the aggregate chip performance is much better than conventional designs with an equivalent number of transistors. Cyclops uses a processor-in-memory (PIM) design where main memory and processing logic are combined (self-contained) into a single piece of silicon. Large, scalable systems are built with a cellular approach using Cyclops as a building block, with the cells interconnected in a regular pattern through communication links provided in each chip.

[0038] In a preferred embodiment shown in Figure 4(b), in the Cyclops design depicted for networking applications, there are eight (8) processor units or "Quads" 100 included with each Quad further connected with internal memory to the embedded shared memory (SRAM) 110 and connected to an on-chip switch fabric 120 which may be an on-chip cross bar switch, or a packet switch fabric, etc. Thus, in one embodiment, the self-contained processor-based subsystem 150 component provides 32 threads of execution, up to 128 kB of local RAM 130 and 512 KByte of shared SRAM 110. It is understood that other designs are possible including 64-bit high-end versions for scientific/engineering applications. In this design, many processing tasks may be broken down into many threads, running concurrently, to provide true multithreading capability. More particularly, as shown in Figure 5, the multiprocessing approach adopted by in the Cyclops architecture, includes many simple cores 125 forming a processor cluster 100' each with a reduced, but general purpose, instruction set of about 40 RISC-like instructions. As shown in Figure 5, each processor core 125 of a cluster 100' has its own register file 126, arithmetic logic unit (ALU) 127, and instruction sequencer 128. In the embodiment depicted, the processor cores 125 have a single-issue architecture with a simple, four stages deep pipeline. Four cores share a local SRAM 130, for storing their stack and local variables, and parts of packets that need to be processed, such as header fields, and may function effectively as an information "cache" device although without any of the usual attributes of a processor data cache. In the embodiment depicted, two four-processor clusters 100' share an instruction cache (I-cache) 131 having a bandwidth for the processors 125 sufficient to prevent instruction starvation and accommodating most working sets of the processor without causing cache trashing and increased instruction miss rate. It is understood that each processor core 125 comprises a thread group and may be connected via the I-cache 131 in order to perform in a multi-threading capability. The more sub-processors (thread groups) the better the overall processor core will operate in terms of faster processor cycle time and reduced cache demands/latency. Exemplary embodiments target 2 - 256 sub-processor groups, with a preferred embodiment of 32 as described herein. However, it is understood that the present invention thus provides true simultaneous, multi-threading, multi-processor design, not limited by the number of sub-processors.

[0039] The small instruction set and simplicity of features allow the processor cores to be of minimal size delivering a high ratio of MIPS/mm^2 of silicon area. This makes possible the placement of many cores on a chip of a rather small footprint to exploit thread-level parallelism. Thus, the present invention may be advantageously applied to enable higher integration / board density for lower card assembly cost; and provide enhanced scalability for larger bandwidth applications and processing cores as it is a true "System-On-A-Chip" implementation, allowing for multiple "cores" for plug-n-play system design, enabling greater architecture flexibility. It is understood however, that the processor core is not scalable to reduce gate or transistor counts based on transmission tasks, or computational load, but rather is a fixed design depending upon the application/targeted market. Further, the multiprocessor or subsystem core does not break down tasks and assign them to a DSP or ASIC functional blocks, rather the program code and data packets are processed in multiple subprocessors (thread groups) each with an equivalent memory (e.g., 16kB SRAM for the data cache) and circuits (ALU, register file, etc). These sub-processors within the thread groups, form thread units, which comprise the processor core as a whole attached to the local system or on-chip local bus (for SOC applications).

[0040] In the present invention, each local processor (thread groups, which in turn contain multiple thread units or further sub-processors) are arranged in a cellular organization, such that each processor has N banks of symmetrical on-chip memory (examples: 256KB SRAM in 4x64KB, or 4 or 8 MB eDRAM in nx512KB blocks), each bank being addressable to each local processor group (thread group) via the crossbar switch. The separate on-chip memory of either SRAM or eDRAM is provided to handle continuous address space to all the sub-processor cores (or thread groups). The integrated 16 KB SRAM memory (one per thread group) is accessible by all the processor threads on the chip.

[0041] In the more detailed view of Figures 5 and 10, the multiprocessor SoC design according to the invention comprises a storage area network (SAN) processor architecture 150' capable of handling network packet communications functions according, but not limited to the following protocols: Fibre Channel 201, Infiniband 202 and Gb Ethernet 203. As shown in Figure 5, the network processor SoC design 150' includes embedded banks of memory 160 for storing data packets, connection information, and programs. Usage of embedded memory (SRAM or DRAM) is advantageous, as significant amounts of memory may be placed on a chip without excessively increasing its size. In addition, embedded memory has short and predictable access times, which can be accounted for in the time budget for the processing of single packets and offers significant performance advantages as compared to conventional off-chip memory as the overall traffic is reduced on the internal interconnect, resulting in fewer resource collisions, reduced performance degradation and power consumption. In addition to storing data, current control, status, and routing information in the embedded memory 160 is maintained. As some applications may require memory requirements exceeding the available on-chip memory, the SoC network processor architecture employs off-chip DRAM (not shown) connected via a high-bandwidth DDR memory interface 165. The external DRAM may store statistics, archival information, as well as provide congestion buffering.

[0042] In the SoC network processor 150' of Figure 5, most of network communications protocol functions are implemented programmatically. However, highly time-critical functions at the lower level of the network protocol are implemented via hardware accelerators. Hardware accelerators handle low-level protocol tasks, such as data encoding/decoding, serialization/deserialization, link management, and CRC and checksum calculation. These tasks are performed on every byte of the transferred packets and would be very computation expensive if implemented in software. The hardware implementation of these functions are thus provided as hardware accelerators implemented in network interfaces 175 for Fibre Channel and Gigabit Ethernet, and a network interface 185 for Infiniband, each requiring only a small silicon area and interfacing with respective Infiniband and Fibre Channel communication links 190, 195.

[0043] Further as shown in Figure 5, the SoC network processor design 150' includes an internal interconnect comprising a crossbar switch 120 that interconnects processor clusters 100', shared memory blocks 160, an external memory interface 165 for external DRAM memory access, and network protocol layer hardware assist devices 175, 185. In an exemplary embodiment, the crossbar switch 120 has 64-bit data paths and provides several words worth of pipelining and token signaling to avoid data overflows. The processor Quad share a port to the crossbar 120, so a crossbar with 16-ports, for example, is sufficient to interconnect up to a 32-processor system. It is understood however, that the crossbar switch 120 may be replaced with a pseudo-crossbar, a bus, a switch, or other such interconnect as may be appropriate, as will be described herein with respect to Figure 8.

[0044] As mentioned generally herein, the SoC network processor architecture is cellular, i.e., it enables the design to be custom scaled depending on the application requirements. For example, endpoint functionality of the Fibre Channel requires less computational power than the more complex TCP/IP termination with iSCSI protocol conversion to Infiniband. In the present invention however, the number of processor cores or clusters 100' and embedded memory blocks may be easily adapted to the application requirements without making significant design changes.

[0045] Figure 6 depicts a first embodiment of an SoC Network Attached Processor 200 employing the self-contained multiprocessor subsystem 150' of Figures 5 and 10. The CyclopsE available from assignee IBM, is one possibility for use as the subsystem 150'. In the embodiment of Figure 6, the subsystem 150' is connected to a processor local bus 210 which may comprise, e.g., a SoC standardized processor- local bus (PLB) such as ARM AMBA (Advanced Microcontroller Bus Architecture), MIPs (Microprocessor Interface Program), the open standard CoreConnect, AHB (Advanced High-Performance Bus), etc. via a common macro (e.g., a PLB connector macro), enabling a true plug-n-play system on a chip (SOC) to a multi-source bus architecture.

[0046] It should be understood that the embodiment of the Network Attached Processor 200 depicted in Figure 6 and 10 implements a PowerPC or other like processor 225 for providing computational capability in the SoC subsystem. Equivalently, a PPC440 may be replaced with another PPC core, a MIPS core, or other such microprocessor as selected by SoC integrator. Likewise, other components depicted in the Figure 6 including SRAM 215, DDR controller 218, PCI-X bridge 222, direct memory access DMA device 226 DMA controller 228, on-chip peripheral bus (OPB) 240 for interfacing with external components via one or more I/O interface devices 245. A Medium Access Control (MAC) protocol device 250 is additionally employed to provide the data link layer for an Ethernet LAN system, processor core timers 233 and interrupt controller 235 may be present or omitted in accordance with selections made by architect/integrator of a specific SoC.

[0047] Figure 7 illustrates a second embodiment of the System-on-Chip (SoC) network attached multiprocessing system 300 according to the invention. As in Figure 6, the SoC multiprocessing system 300 of Figure 7 comprises the processor (e.g., a 440 core), a local processor bus (PLB) 210, on-chip peripheral bus (OPB), and a number of components, such as SRAM, DDR controller, PCI-X bridge, and DMA controller, however includes an OPB bridge 229 interfacing with the OPB bus 240. The processor bus or PLB 210 is a SoC standardized processor local bus such as AMBA, MIPs, CoreConnect PLB, AHB, etc. One of the components connected to the PLB 210 is a processor based subsystem 350 described in greater detail hereinbelow with respect to Figure 8. The elements depicted in Figure 7 are exemplary and non-limiting. For example, PPC440 can be replaced with another PPC core like PPC 405 or PPC440, or ARM or MIPS processor cores, or other such microprocessor as selected by SoC integrator or may include completely novel cores without limiting the main scope of this invention. Likewise, other components listed here (or any other component from the SoC library) may be present or omitted in accordance with selections made by architect/integrator of a specific SoC. For instance, as shown in Figure 7, devices provided for interfacing with the On-chip Peripheral bus 240 may include, but is not limited, one or more of the following: a RAM/ROM Peripheral controller 245a, an external bus master 245b, a UART device 245c, an Inter-IC bus (I2C) interface 245d, general purpose I/O interface 245e and a gateway interface 245f.

Thus it is understood that there is enabled multiple-chip configurations.

[0048] Figure 8 depicts a self-contained processor-based subsystem 350 according to further embodiment of the invention. This subsystem is integrated as a component in a SoC network attached processor system such as depicted in Figures 6 and 7 and is connected to a processor bus 210 via a PLB bridge which can be a common macro in the ASIC library. The processor based subsystem 350 comprises one or multiple processor clusters such as the processor cluster 100' of Figure 5, one or more local memory cells for storing data and/or instructions and local interconnect means implemented as a separate bus, fabric, crossbar switch or other interconnect means 120. In the preferred embodiment, the multiprocessor subsystem 350 comprises a PLB bridge macro component 410 for communicating over the SoC network processor bus 210, however it is understood that any other bridging macro can be selected to enable data flow between the processor based subsystem 350 and the SoC bus 210. The processor bus 210 is a separate bus, switch or interconnect means used in System-on-Chip assembly for connecting a processor and components.

[0049] Separation of the subsystem and the processor buses 210 (Figures 6 and 7) is advantageous in that: 1) subsystem traffic is separated from the PLB traffic, avoiding bandwidth contention; 2) only traffic between the subsystem and the SoC system on global standardized bus is the interface traffic (data receive and send); 3) the subsystem bus/switch interconnect fabric is designed to offer optimized MP fabric for implementing high performance solution, without requirement to accommodate standardized components and connection protocols in a SoC system. In this way, a SoC solution may benefit from both worlds: the multiprocessor (MP) fabric can be optimized for MP high performance, and all the standard existing components from the SoC library can be used.

[0050] The subsystem 350 including interconnecting bus/ switch/fabric 120 is particularly connected to a processor bus 210 using a bridging component 410 which adapts for different speeds, data widths, signals and signaling protocols between two communication systems, in the way existing bridges perform, e.g., PLB-to-OPB bridge, or PLB-to-PCI-X. Implementing an interface to a standardized processor local interconnect such as PLB or AMBA enables integration of this new component into SoC component library. A possible implementation of this bridge component 410 is shown in Figure 9. The purpose of this bridge macro 410 is to translate / adjust control signals, data width, operating frequency and address space between the SoC processor bus 210 and the processor based subsystem local bus 120. Preferably, bridge macro component 410 implements data buffering for data coming to and out of the processor-based subsystem module, and may include DMA controllers for subsystem and PLB. The configuration and status registers may be implemented as memory-mapped registers in the subsystem address space. The configuration registers are set by the processor-based subsystem 350, and it also reads the status of the bridge 410. This module can also include settings to select between various data width on the SoC processor bus (e.g., to set operation mode to work with 64 or 128 PLB), and/or to support various modes of operation, e.g., line and burst data transfers. The SoC address space and the subsystem address spaces may, but not necessarily have to share the same address space.

[0051] The bridge macro 410 of Figure 9 particularly functions on the PLB bus as a PLB slave 420a and as a PLB master 420b. As a PLB slave, it implements read and write requests from the SoC processor for getting a piece of data from the processor based subsystem 350. During a read request for a data in the memory in the processor based subsystem, the bridge receives a read request from the PLB 210, resolves address and generates read request for the processor based subsystem bus/fabric/switch 120. It buffers read data from the processor-based subsystem 350, and transfers data to the PLB 210 in the width and speed specified by the PLB bus 210. During a write request for a data in the memory in the processor based subsystem, the bridge buffers data from PLB 210 for write request, resolves address for the memory bank in the processor-based subsystem, and transfers data to the proper memory bank in the processor-based subsystem 350, as specified by its bus/fabric/switch 120.

[0052] Conversely, when functioning as a PLB master 420b, it can but must not implement a DMA controller for transferring data from and to the processor-based subsystem. In transferring data by the DMA controller from the processor-based subsystem to the DDR memory of the SoC, the controller sets address and signaling for PLB write request, and then transfers data to the DDR memory. During the DMA transfer of data from the DDR to the processor-based subsystem, the macro sets address and signaling for PLB read request, buffers data, and transfers data to the memory bank in the processor-based subsystem.

[0053] In the preferred embodiment, the processor based subsystem 350 comprises an embedded software providing ready-made functionality (personalization) for a specific set of functions. Possible use are network protocol conversion of one network protocol to another, protocol traffic termination, like TCP/IP offload engine, IPSec VPN tunneling engine, network processing for iSCSI, encryption engine, compression/decompression engine, or for multimedia processing, like MPEG en/de-coding, or sound/voice/video processing.

[0054] Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is: